$2.00

VOLUME 2 · 1987

# R/D COMPUTING

VER 23
MAY

Dedicated to TI 99/4A and 9900 Computer Systems

Ryte Data

210 MOUNTAIN STREET,
HALIBURTON, ONTARIO K0M 1S0
(705) 457-2774

WELCOME ABOARD:
We have some new personnel on board. I would like to welcome Joe Warder of Regal Capital, Bill Weiler and Michael Jarman of Helix Circuits, as well as Henri Schlereth and Judi Beckett for Ryte Data South.

Growing from start-up in publishing to software publishing to hardware on a cash only basis has been VERY tough. Here only the tough survive. In the TI market, most of the players have left for greener pastures. The companies that remain are tenacious survivors and visionaries who BELIEVE in the TMS 99xx technology. They do deserve your support.
Then again, all of YOU have proven to be supporters. The ones we need to reach are 'out in the cold'; owners who do not belong to users groups, owners who have closeted their machines, owners who have yet to upgrade their "home" computer and others who are not aware of the power contained in their consoles.

INDUSTRY NEWS:

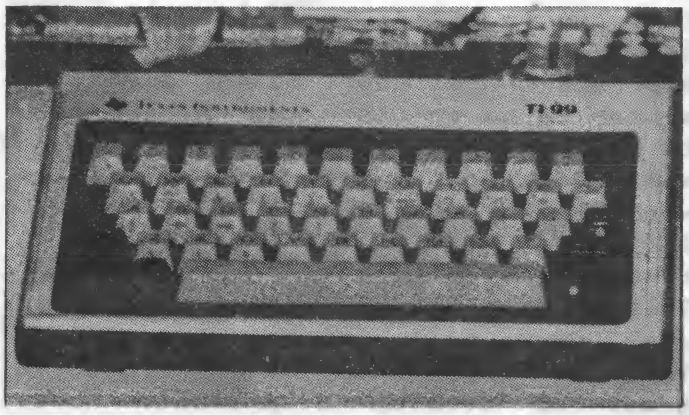
The announcements of the new machines from Apple and IBM have created a wave of optimism in the computer industry. This is good news for computer users, support companies and manufacturers. With all the high powered hardware being introduced, the choices open to computer users are increasing.

Many believe that the MacIntosh SE and the MacIntosh II are THE products to watch. The Mac II incorporates far reaching technology, expansion slots, IBM 8088 co-processors and true cross system connectivity (one of the new buzzwords).
Stay tuned, this is the arena.

This issue is a collection of various hardware projects, tips and the like that have some very good results. We have found that the improvements are both valuable and quite noticeable. With the extended memory maps and new card based projects, it is possible to transform the TI on a user basis!

We would like to encourage owners to attempt this type of electronics modification. With the price of spare consoles at an all time low, there is relatively little danger of doing irreparable damage. Get out your soldering iron and parts, try some projects and get the results.

Of particular note is the very simple resistor change suggested by Bob Lawson. This two cent change gives a clearer video display... something else that TI "should" have done in the basic design.



HARDWARE GROUP:

The response to the hardware SIG co-ordinated by Bob (Tony) Wagner is moving right along. Over three dozen people have responded to the call. I believe this group has more talent than even TI could muster. Several projects are in motion which will have far reaching effects over the next

several months. Hardware inputs from the group promise to give all TI 99/4A owners some astounding results.

As with anything that contributes to the support of our orphan, your support is both welcome and ABSOLUTELY essential. Nothing occurs in a vaccum. Even more to the point, a little participation goes a long way in providing the "afterlife" phenomena we witness in the 4A world. One of the major challenges facing a hardware group is that of financing the work done. This will have a telling influence on the progress.

The top priority project (in my book) is an extended memory specification. A modification to the operating system would be required - but would allow the 99/4A to access and USE additional memory... the key component for more sophisticated software.

CO-PROCESSORS:

Ahhh! Now this is a timely focus... William Borchardt of the Sun City 99ers writes that he is working on a co-processing system that will allow the removal of the GROMs from the console. This would entail altering the 99/4A clocking system to allow an increase in speed from the machine. The intent is to allow either the GPL approach or a user built operating system to be put into the co-processor to be accessed from the 99/4A in the same fashion that the GROM interface works - only at a higher clock speed. It looks as if the basic design is finally coming together.
This is the type of project that would work very well in a group format. There is always a LOT of work to do in order to develop a project. Monty Schmidt tells me that he has an 8088 'co-processor' project up and running for his university class. Interfaces through the 8k DSR space on the DSR RAM CARD. THIS type of project 'could' provide a true IBM compatibility mode with bus to bus information transfer. Only a large amount of money, time and talent would produce a commercial project. It does sound interesting.

OTTAWA TI FAIRE 1987

Having missed the New Jersey and the Boston TI Faires (sorry folks), we did manage to make the Ottawa show on May 16th at the Merrivale High School. With the holiday weekend, the turnout was a bit light. The number of vendors was down from last year and the number of TI owners also seemed to be shy of last years event.

We set up next to Myarc - with Lou Philips and Walt Howe showing the Geneve 9640 computer. We had our 99AT Expansion Systems on display along with Monty Schmidt's COMMAND DOS (also now shipping).

I can personally verify that Myarcs 9640 is ALL that it is supposed to be and more. In fact, some of our own subscribers and Canadian TI 99/4A owners now HAVE this new machine in their possession. Steve Michelson of the Toronto 9T9'ers walked out with one under his arm. Disk Only Software was selling and taking orders for Myarc's Geneve 9640 computer system.

Myarc is shipping the machine with a full 350 page manual. All software that is specified to work - WORKS. I saw a number of modules downloaded to run. This means that software IS available... and more new software will become available. I also personally know of several programmers who are releasing packages SOON - if not before.

For us the show was important in that we were able to determine exactly how much power the card pulls from the card bus. Very critical in determining the total compatibility of the 99AT Expansion System.

Scott Darling, the SYSOP of GENIE was at the show. GENIE has come to Canada through connection nodes in several cities. Try calling 1(800) 638-9636 for more information. We also met many other TI 99/4A luminaries and supporters.

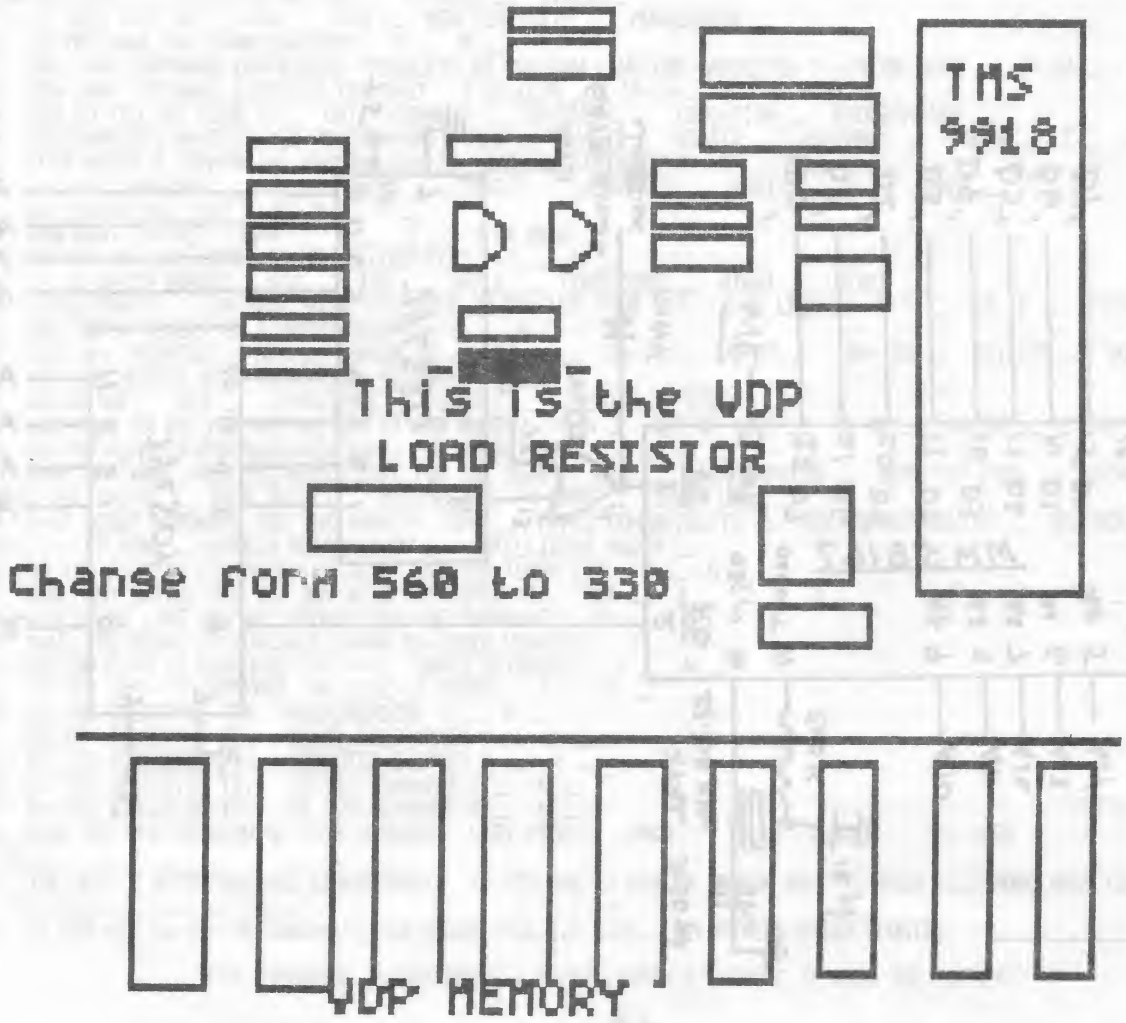
# Improved Video

by, Bob Lawson

In my travels through the Texas Instruments Manuals, specifically the TMS-9918,28,29 Manual, I read,"The load resistor (RL, pin 36 to ground) defines the sharpness of the edges on the video signals. A lower resistor value gives faster fall times and a sharper picture." Hmmm! I don't remember any 330 ohm resistors.

Well, I pulled out the "TI Console and Peripheral Manual, and sure enough, R212 pin 36 to ground was 560 ohms per the schematic. The next step was to check out a console, and well you guessed it, R212 was 560 ohms, not 330 ohms as recommended in the TI Manual!

Next step was to try some different value resistors, 330 ohms seems to be about the best common value resistor to use. I wonder why TI chose to use 560 ohms. I did find one old TI Manual which recommended 390 ohms (1979), but they're sometimes hard to find in 1/4 watt. This 30 cent change gives about a 40%, that's right, I said 40% improvement in the picture. The improvement is so good, you'll wonder where the WHITE SHADOWS WENT.



## Adding a Real Time Clock to the 8K DSR Card

John Clulow (GEnie address J.CLULOW)
(419) 874-8838

This project adds the National Semiconductor MM58167 clock to the 8K DSR RAM card. Before you attempt to add the clock, you must complete the non-DSR 8K addition. Make sure the card is working properly. Double check the wiring of the data lines in the 74LS245. D0 - D7 must be wired to the edgecard connector exactly as shown in the schematic. While the memory will work with any order of data line connection, the clock will not.

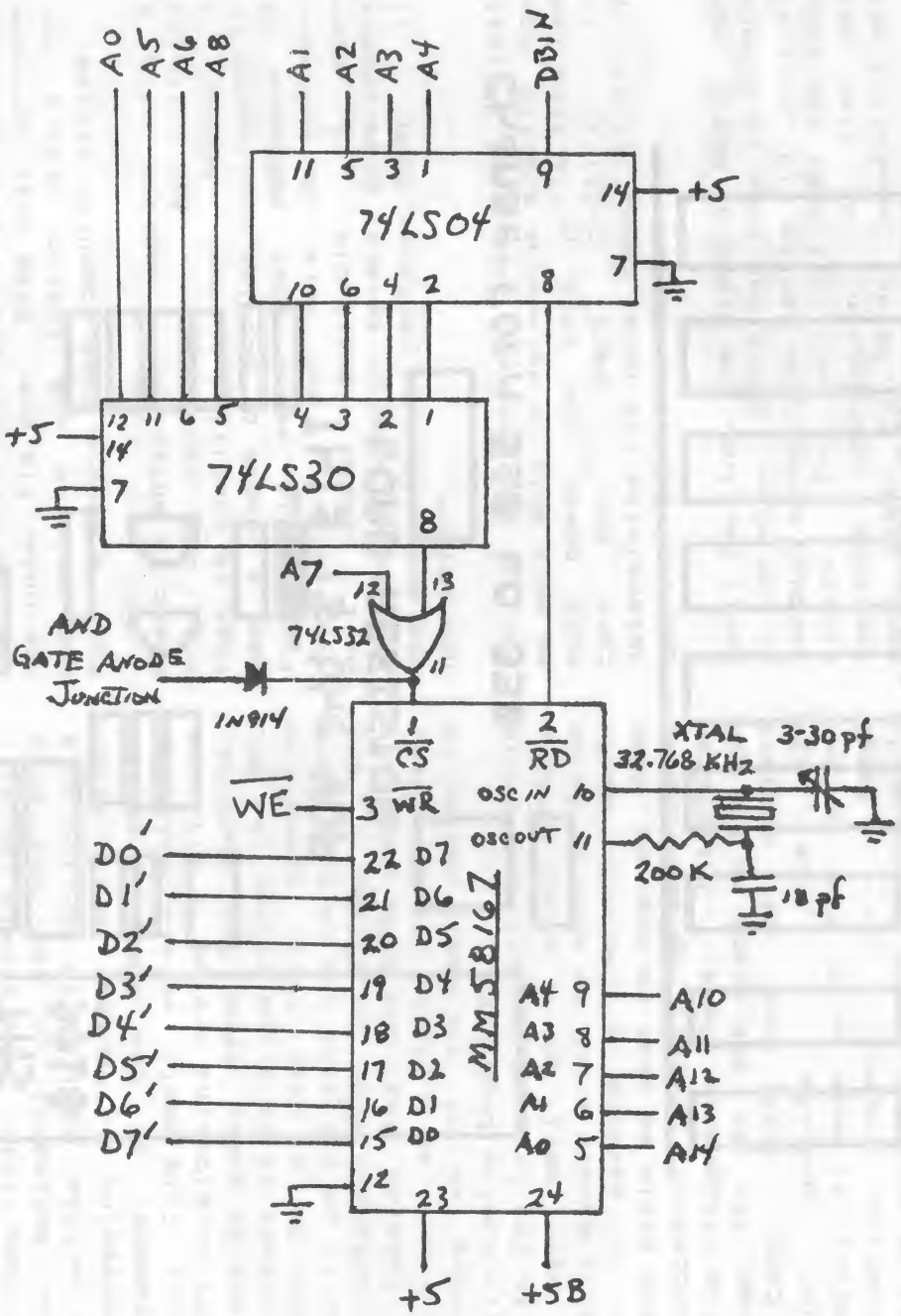
To complete this project, you will need a 74LS04, 74LS30, MM58167 clock, 32.768 K Hz microwatt crystal, 1N914 diode, 200K ohm resistor, and two small capacitors. One capacitor is fixed and should be around 20 pF. The other is a variable capacitor with a range of 3 to 30 pF. Digi-Key (800-344-4539) is one source of these parts.

Wire the circuit as shown in the diagram. The address lines and DBIN need not be connected to the edgecard connector; you can refer to the 8K card schematic and use appropriate pins on the 6264 or 138, for example. For D0' to D7', use the 74LS245 pins 2 through 9, and make sure the corresponding input pins (18 - 12) are properly connected to the edgecard connector. The anode end of the 1N914 diode should be connected to the junction of the anodes of the two diodes used in the AND gate you made in the non-DSR RAM project (see step 6). It could also be connected to pin 15 of the 74LS368 or pin 19 of the 74LS245.

When you purchase your clock IC, obtain a data sheet. Here is a partial list of addresses used in working with the clock:

| | | | |
|---|---|---|---|
| 8680 .001 sec | 8686 minutes | 868C day of month | |
| 8682 .01 sec | 8688 hours | 868E months | |
| 8684 seconds | 868A day or week | 86A8 status | 86A4 reset |

Software for the clock is available from the GEnie library.



TI-99/4A OWNERS SURVEY DATE:_____

PLEASE USE A DARK COLORED FELT PEN, THANK-YOU
YOU MUST SELECT ONLY 1 ANSWER, NO EXCEPTIONS.

A1(2) WHAT IS YOUR SEX? M __ F __
A2(5) WHAT IS YOUR AGE GROUP? 17 OR UNDER __ 18-24 __ 25-34 __ 35-44 __ 45 AND OVER __
A3(5) HOW MANY YEARS OF EDUCATION? 11 OR LESS __ 12 __ 13-15 __ 16 __ 17 AND OVER __
A4(5) YOUR OCCUPATION? STUDENT __ BLUE COLLAR __ WHITE COLLAR __ PROFESSIONAL __ RETIRED __
A5(2) DO YOU USE A COMPUTER AT WORK? Y __ N __
A6(8) WHICH BRAND? N/A __ IBM __ APPLE __ DEC __ SPERRY __ NEC __ DONT KNOW __ OTHER __
A7(3) HOW MANY TI-994A OR 4As DO YOU OWN? 1 __ 2 __ 3 OR MORE __
A8(4) HOW MANY PESystems DO YOU OWN? 0 __ 1 __ 2 __ 3 OR MORE __
A9(7) WHICH MEMORY EXPANSION DO YOU HAVE? NONE __ TI __ CORCOMP __ FOUNDATION __ MYARC __ MECHATRONIC __ OTHER __
A10(6) WHAT SIZE? N/A __ 32K __ 128K __ 256K __ 512K __ 1024 __
A11(6) WHICH DISK CONTROLLER CARD DO YOU OWN? NONE __ TI __ FOUNDATION __ MYARC __ CORCOMP __ OTHER __
A12(4) HOW MANY 5-1/4" DISK DRIVES DO YOU OWN? NONE __ 1 __ 2 __ 3 OR MORE __
A13(5) CONFIGURATION? N/A __ SS/SD __ DS/SD __ DS/DD __ DS/QD __
A14(2) DO YOU OWN A HARD DISK SYSTEM? Y __ N__
A15(4) WHATS THE SIZE? N/A __ 10 OR LESS MEG __ 20 MEG __ 30 OR MORE MEG __
A16(2) DO YOU OWN A MODEM? Y __ N __
A17(4) INDICATE HIGHEST BAUD RATE? N/A __ 300 __ 1200 __ 2400 __
A18(10) WHICH PRINTER DO YOU OWN? NONE __ TI __ EPSON __ PROWRITER/NEC __ OKIDATA/C ITOH __
STAR MICRONICS __ COMREX/TOSHIBA __ BROTHER/CANNON __ JUKI/CITIZEN __ OTHER __
A19(2) DO YOU OWN A MONITOR? Y __ N __
A20(2) ARE YOU USING A TV? Y __ N __
A21(10) WHICH MONITOR BRAND? N/A __ TI __ AMDEK __ SANYO/TAXAN __ NEC __ TATUNG/SAMSUNG __
MAGNAVOX/SONY __ EPSON/PRINCETON __ HITACHI __ OTHER __
A22(4) IS YOUR MONITOR? N/A __ RGB __ RGB/COMPOSITE __ MONOCHROME __
A23(2) DO YOU HAVE THE GRAM KRACKER? Y __ N __
A24(3) HOW MANY SOFTWARE CARTRIDGES (MODULES) DO YOU OWN (INCLUDE CASSETTE) ? 9 OR LESS __ 10-24 __ 25 OR MORE __
A25(3) HOW MANY SOFTWARE DISKS DO YOU OWN? 9 OR LESS __ 10-24 __ 25 OR MORE __
A26(4) HOW DO YOU USE YOUR TI? ENTERTAINMENT __ BUSINESS __ EDUCATION __ PROGRAMMING __
A27(7) PROGRAMMERS, WHAT DO YOU PROGRAM IN? N/A __ BASIC __ XBASIC __ ASSEMBLY __ FORTH __ C __ OTHER __
A28(2) EVER WRITE A COMMERCIAL PROGRAM FOR TI (INCLUDE FREEWARE)? Y __ N __
A29(7) IN WHAT LANGUAGE? N/A __ BASIC __ XBASIC __ ASSEMBLY __ FORTH __ C __ OTHER __
A30(2) DO OTHERS USE YOUR TI? Y __ N __
A31(4) HOW MANY OTHERS? N/A __ 1 __ 2 __ 3 OR MORE __
A32(2) DO YOU OWN ANY OTHER BRAND OF COMPUTER? Y __ N __
A33(6) IF SO WHICH BRAND? N/A __ IBM __ APPLE __ COMMODORE __ ATARI __ OTHER __
A34(4) APPROXIMATELLY HOW MANY HOURS PER WEEK DO YOU USE YOUR TI? 4 OR LESS __ 5-9 __ 10-14 __ 15 OR MORE __
A35(2) ARE YOU A MEMBER OF A USERS GROUP? Y __ N __
A36(8) HOW MANY MEMBERS? N/A __ UNDER 33 __ 34-66 __ 50-74 __ 75-99 __ 100-150 __ 151-199 __ OVER 200 __
A37(2) ARE YOU MEMBER OF A COMMERCIAL NETWORK? Y __ N __
A38(5) WHICH ONE? N/A __ COMPUSERVE __ SOURCE __ GENIE __ OTHER __
A39(6) HOW MUCH IN DOLLARS DO YOU USE IT PER MONTH? N/A __ $29 OR LESS __ 30-49 __ 50-74 __ 75-99 __ 100 OR MORE __
A40(2) DO YOU PLAN TO PURCHASE (OR HAVE) THE MYARC 9640? Y __ N __
A41(9) WHAT SOFTWARE AREA YOU WOULD LIKE FOR THE 9640? N/A __ CAD/GRAPHICS __ DESK-TOP PUBL __ DATABASE __
WORD/PROC/MULTIPLAN __ TRUE BASIC __ ASSEMBLY __ C-LANG __ OTHER LANG __
A42(8) WHAT ABOUT HARDWARE FOR THE 9640 ? N/A __ APPLE COMPATIBLITY __ IBM COMPATIBILITY __ RGB MONITOR __ 3.5" MICRO
DISKETTE __ CD ROM __ SPEECH RECOGNITION __ INTELLIGENT MODEM __
A43(2) DO YOU PLAN TO PURCHASE (OR HAVE) TRITONS TURBO XT? Y __ N __
A44(8) HOW MANY EXTRAS DID YOU BUY WITH THE TURBO-XT? N/A __ 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __
A45(2) WOULD YOU LIKE AN 'AT' UPDATE FOR THE TURBO-XT? Y __ N __
A46(3) HAVE YOU PLANS TO PURCHASE CARDS BY OTHER VENDORS? Y __ N __
A47(2) DO YOU PLAN TO PURCHASE (OR HAVE) RAVES KEYBOARD? Y __ N __
A48(2) DO YOU PLAN TO PURCHASE (OR HAVE) A MOUSE? Y __ N __
A49(2) DO YOU SUBSCRIBE TO: MICROPENDIUM Y __ N __
A50(2) DO YOU SUBSCRIBE TO: SMART PROGRAMMER Y __ N __
A51(2) DO YOU SUBSCRIBE TO: COMPUTER SHOPPER Y __ N __
A52(2) DO YOU SUBSCRIBE TO: GENIAL TRAVelER Y __ N __
A53(2) DO YOU SUBSCRIBE TO: UG PUBLICATIONS Y __ N __
A54(5) WHAT IS YOUR OPINION OF THIS SURVEY? VERY POOR __ POOR __ OK __ GOOD __ VERY GOOD __

AC$(3) THE NEXT 2 QUESTIONS ARE DEMOGRAPHIC. IF YOU ARE IN USA OR CANADA WHAT IS YOUR TELEPHONE AREA CODE. ALL OTHERS ENTER
CITY: _____
ZC$(3) IF YOU ARE IN USA OR CANADA PLEASE ENTER YOUR ZIP CODE. ALL OTHERS ENTER COUNTRY: _____

FOR COMMENTS, PLEASE WRITE A BRIEF LETTER & ENCLOSE IT WITH THE SURVEY.

C99 PART THREE

by Ron Albright Jr.

Last time we touched on what c99 is, and what files come on the disk - plus what some of the more important ones do. This time we'll actually do some code. As we progress, we will stress some sort of style in how we enter programs. I am no expert on style (or c99 for that matter), but since c99 is so free-form and has no line numbers to follow, it can be very difficult to read programs if you don't follow some rules. These rules are not universally agreed upon, but we'll try to develop some sort of easy to read style of our own. I will make a few assumptions to start:

First, I will assume that you have a single drive system with only single sided capability.
Second, I will assume that you have a basic understanding of the Editor/Assembler package, i.e. you know how to use the Editor and run programs out of either Option 3 or Option 5.

I will further assume, that you have assembled at least one source code file with E/A. If these assumptions are incorrect, let me know and we'll touch on the Editor Assembler more next time. Let's get started.

Take a clean disk and copy the following c99 files onto it:

| | | |
|---|---|---|
| CSUP | D/F 80 | 12 SECTORS |
| PRINTF | D/F 80 | 14 SECTORS |
| UTIL1 | PROGRAM | 33 SECTORS |
| UTIL2 | PROGRAM | 33 SECTORS |
| UTIL3 | PROGRAM | 29 SECTORS |

Next, from the Editor/Assembler disk, copy these files to the same disk:

| | | |
|---|---|---|
| ASSM1 | PROGRAM | 33 SECTORS |
| ASSM2 | PROGRAM | 20 SECTORS |
| EDIT1 | PROGRAM | 25 SECTORS |

If my addition is correct, that gives you 199 sectors on our work disk. Now we are ready to proceed. Keep our work disk in the drive and insert the Editor/Assembler cartridge. From the menu, load the Editor and go into the Edit mode.

Type in this program:

```
/* c99 The smallest c99 program */

main() /* a comment */

{
    /* we aren't going to do anything!
*/

}
```

Congratulations! You have just entered your first, valid c99 program. Let's look at it. The first line is nothing more than a "REM" statement. Instead of REM, c99 recognizes anything enclosed within "/* */ as a comment and ignores it when compiling. You can put anything between these comment delimiters, and it will survive compiling without error. Use them frequently as you program. As we mentioned, c99 programs are difficult to read at best and REM statements are useful to remind yourself, as well as others reading the program, what you had in mind. As shown on the next program line, the /* can be used on the same line as compilable code, so comment each step of your code for clarity.

A routine called "main" is required somewhere in each and every c99 program. Typically, it is the first block of code, sets things up and calls the other routine(s) to take over. When the compiler sees "main()" (or anything else with the "()" after it - like "first()", "setup()" - it labels this as a function. A function is similar to a subroutine in Extended Basic. A string of functions make up a program. They are just like using "SUB routine" in XB. It is run when its name ("main", "first", "setup") is 'called'. The "main" routine is run whether it is called or not called (guess that is why they call it "main"). More on this later.

For now, think of c99 as simply a series of "calls" to blocks of modular code called functions with each functions labeled with "name()".
Each function is enclosed with a pair of braces - it starts with an open brace ( { ) and ends with a closed brace ( } ). This tells the compiler where this block of code starts and ends. Everything within those braces is part of that function. In our first program, the only thing in the main function is a "REM" statement, so it will 'do' nothing. It is compilable though. A function may include a call for another function. Look at this example:

```
main()
{
     doit();
}

/* doit doesn't do anything! */

doit()
{
/* see!  Nothing here to do! */
}
```

This time, main calls up the second function, "doit" which, also, doesn't do anything. You can see how programs are built. Typically, (but not necessarily) the main function will include all the calls to the functions that make up a whole c99 program. It's like having an XB program that is nothing more than a series of "GOSUB"s (really a series of "CALL SUB" routines). Each function call does its own task and returns control back to the main, or controlling program. The good c99 program will break large programs into smaller ones and write a function for each. If a function can stand alone (has nothing in it unique to a single program) the programmer eventually develops a 'toolbox' of useful small routines (functions) that can be combined in different ways to solve problems. That is just one of the beauties of c99.

So, let's compile this program. After typing it in, hit FCTN 9 twice, get the EDITOR menu and elect to save it to disk. Your main work disk should have plenty of room without disk swapping. After saving to Disk 1, hit FCTN 9 again to get the main E/A menu. Chose Option 5 to "RUN PROGRAM FILE'. The three compiler files, which I have renamed UTIL1, UTIL2 and UTIL3 run out of Option 5, not Option 3 (which runs D/F80 files). When you are prompted for "Program Name:" you only have to hit enter since you have changed your compiler files to UTIL1 - 3. The default name for E/A 5 is UTIL1 and those files will be loaded automatically. You see why I renamed them. You will then be prompted by the c99 compiler (prompts will vary depending on which version of c99 you use) for an input file name. Type "DSK1.filename" (filename being generic for whatever you called the file you typed in and saved to disk). You will then be prompted for an output file name. Call it "filename/C", just to remind yourself that it is a compiled file. Then, hit ENTER and you are off and running. The compiler will flash each function name on the screen as it is compiled to show you where you are in the program. You should see only "main" if you are compiling the first program and "main" then "doit" if you are compiling the second program. If an error is encountered, you will be told. We'll assume that you typed these short routines in without error for now. It shouldn't take long. You are told to press enter to continue after the compiler is done.

Now what? If you catalog your disk now, you should see the initial source code file you typed in and saved with a second filename called "filename/C". Both should be D/V80. You have one more step to do before the program can be run. What the compiler produced was assembly language source code. Like all source code, it has to be assembled. Get to the main E/A menu and choose Option 2, Assemble. When asked to "Load Assembler?" hit "Y" to load the E/A assembler files (ASSM1 and ASSM2). Since we placed these on the work disk, they should load right in without swapping disks. You are

then prompted for the "Source File Name". Type in "DSK1.filename/C" (NOT the program you typed in and saved, but the compiler's output filename). For an "Output File Name", I use "DSK1.filename/O" to let me know this is object code. Then hit ENTER for each of the next two assembler prompts ("List File Name" and "Options"). The assembler should start right up and finish with the assembly process. Now, catalog your disk again. You should see a third file added - "filename/O". This time, it is not D/V80, but D/F80... assembly language OBJECT code. You have produced an assembly language program. How do you run this 'do nothing' program you have written? Go back to the main E/A menu. Choose Option 3 from the menu. When asked for "File Name", type in "DSK1.filename/O". Then hit enter. You get the same prompt again "File Name". This time, type "DSK1.CSUP". This c99 support file MUST be loaded after you load ANY c99 program. Hit enter. When you get the prompt for the third filename, just hit enter this time. When asked for the "Program Name", type in "START". All c99 programs run with the program name START. Your do nothing, super- duper assembly language program should now "run". You then immediately get the "Hit enter to continue" message and you have finished.

Well, how does it feel to have generated an assembly language program just like the "big boys"? Next time, we will do something with a little more substance. We will create a simple menu, which will demonstrate keyboard input and the "printf", "puts" and "getchar" functions. But, for now, I just wanted to go through the mechanics of running the c99 system. Till the next tutorial, get a C book, read the manual that comes with c99 itself, send for the newest version [3.0] of the compiler (pay Clint) and we'll progress further next time.

(c) 1986 Ron Albright

May be reprinted by any non-profit organization without permission.


# QUALITY 99 SOFTWARE

**QS-CONVERTER**

Convert a text file into a running program. Create or modify a program using the powerful editing capabilities of TI-WRITER or QS-WRITER (delete blocks of lines, save or merge blocks of lines, change variable names, etc.). Then use QS-CONVERTER to change the text file back into a running program. 100% Assembly Language.

Reg. $39.95 **Sale ~~$19.95~~ 14.95**

**QS-RAMDISK**

Turn your Foundation 128K card into a true, 127 file, 360 sector, RAM disk with no file size or file type restrictions!

Reg. $39.95 **Sale ~~$29.95~~ 24.95**

**QS-RAMCHIP**

If you can currently access DSKX, then you need QS-RAMCHIP in order for QS-RAMDISK to work. All current programs will still work. Easy to install. (Sold only with QS-RAMDISK.)

Reg. $24.95 **Sale $19.95**

**QS-XREF**

Save hours of your time in writing, modifying, or debugging programs. Produces a full cross-reference listing by line numbers, of all keywords, functions, variables, arrays, subprograms, and line number references, in ONLY 2 minutes. Over 30 times faster than Programming Aids III. Shows where each variable and line number is used. Also shows where each variable is changed. No disk swapping or accessing. 100% Assembly Language.

Reg. $29.95 **Sale ~~$19.95~~ 14.95**

**SOFTKEYS**

Enter commands with only one keystroke! Save time and eliminate typing errors. Define 10 keys to be any commands you want, e.g. SAVE "DSK1. or LIST "PIO" etc. Then, only one keystroke will instantly enter the entire command. A great time saver. 100% Assembly Language.

Reg. $19.95 **Sale ~~$14.95~~ 9.95**

[All programs require disk, 32K, Ex. BASIC.]

Add $4.50 for S&H. Ask for FREE catalog #23R.

For fastest service, call:

VISA **(202) 667-3574** MasterCard

or write:

QUALITY 99 SOFTWARE
1884 Columbia Rd #1021
Washington DC 20009

## Memory Access Indicators
**********

### For 8K DSR Non-DSR Card
**********

8K Card Design By J.Clulow
LED Upgrade ---By S.Tuorto

This project is designed to be easily installed on John's 8K card. When added to your card this upgrade will, at a glance, tell you exactly what memory block your computer is accessing. It displays the entire memory map from >0000->FFFF. This is done using 8 LEDs each connected to 1 8k block. The display board has been designed to be removable (disconnects easily) from the 8K card. The completed assembly can be easily affixed to the rear of the computer console. NO drilling or modification of the console is required!

The idea for this project was from an article in the West Penn 99er's newsletter. The article was about drilling and installing 4 LEDS in the console to show 32K memory expansion access. The original article was by John Willforth of the West Penn 99er's. [Editor's note: John Willforth is working on having a PEB card designed to make these, and other, projects easier. Write if you are interested in a card based kit!]

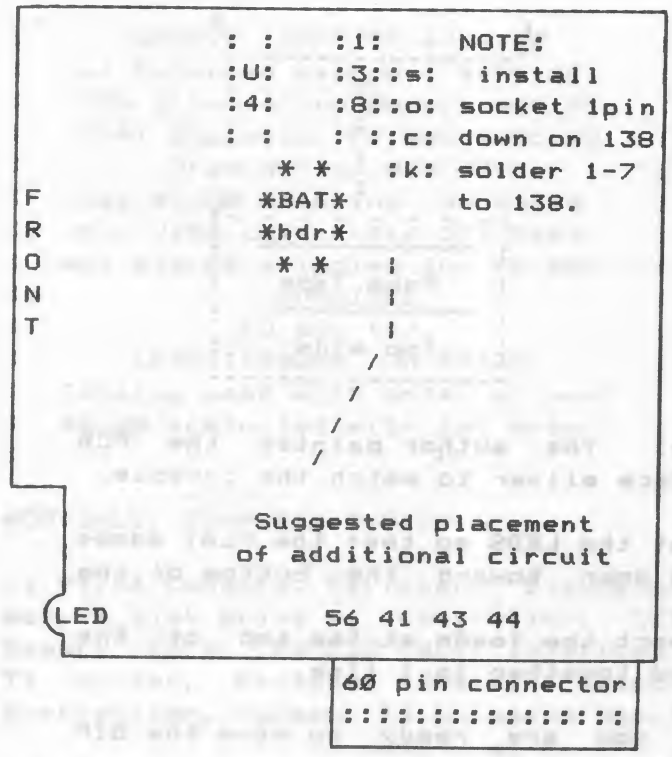
With exception of the 74LS138 chip, all parts are available at Radio Shack.

PARTS LIST
---

| QTY | PART NUMBER | DESCRIPTION |
|---|---|---|
| 1 | 74LS138 | I.C. |
| 8 | 276-033 | Fresnal LED |
| 1 | 271-017(.5w) | Resistor 330 |
| 2 | 276-1998 | 16 pin socket |
| 1 | 276-1980 | DIP Header |
| 1 | 3-ft. 9cond. | Ribbon cable |
| 1 | 276-149 | P.C.B. |
| 1 | 64-2343 | DBLsided tape |

Misc. Silver paint-wirewrap wire-tape

8K (16k) DSR-NON-DSR PCB




Connect the 74LS138 as follows:

Pin #1 to 41(A2), Pin #2 to 44(A1), Pin #3 to 43(A0), Pin #4 to 56(mem.a) of the 60 pin connector.

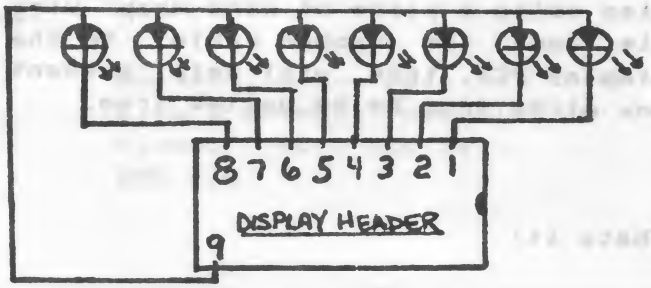
Pin #5 and #8 to Ground(-) Pin #6 and #16 to +5v

Pin #7 to header socket #8.

Next on the header socket solder pins 1-7 to the 74LS138 (15-9). Same method as used with the dip sw. 1 pin down and connect it to the IC.

Finally connect Pin #9 of the header socket to the 330 ohm resistor to +5v

This completes the modification of the 8K board.



```
              DISPLAY
              -------
           1 2 3 4 5 6 7 8

      L                     R
      ----------------
                 :
                 :
                 :
                 :
      :- - - - - - - - - -:
      : ~~~~~~~~~~~~~~~~  :
      :    Foam Tape      :
      : ~~~~~~~~~~~~~~~~  :
      :    top side       :
      ----------------
```

Note: The author painted the PCB surface silver to match the console.

Mount the LEDS so that the FLAT edges face down toward the bottom of the PCB.
Connect the leads at the top of the board together in 1 line.

Now you are ready to wire the DIP HEADER as follows.

Now using a ribbon cable connect one end to the header pins 1-8 and last PIN #9 (+5v), if possible use a different color for this Pin (red).

The other end of this cable is connected to the Display PCB. Left - right numbers 1,2,3,4,5,6,7,8 and finally connect pin 9 to the top line of the LEDS.

Now affix piece of double sided foam tape to the surface of the PCB but do not remove adhesive cover from the side facing you.

Cover the rear side of the board at the LED connections with a piece of electrical tape to insulate them from possibly shorting against some metal object.

Also using a piece of wire wrap wire tie down the ribbon cable to the display PCB, this will help prevent any wires from being pulled free.

Thats it!

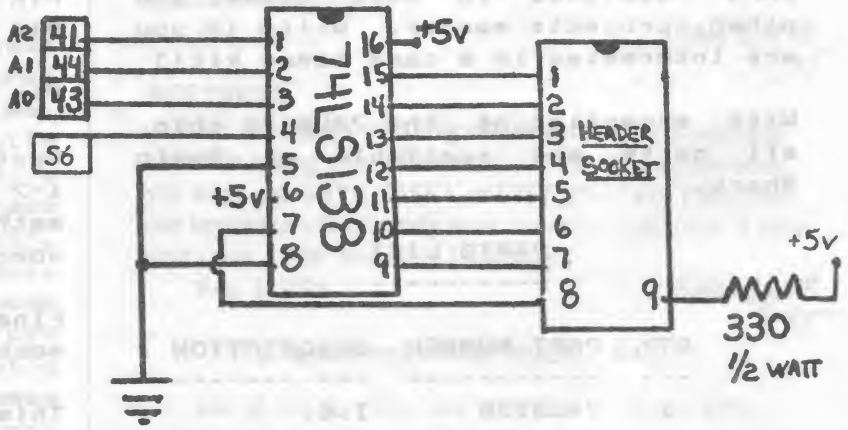
As a precaution, with the cable plugged into the socket you should check for any shorts between pin #9 and any other pin on the Dip Header socket.

The author placed the displayboard on the rear surface of the console with the LEDS about 1/4 to 1/2 inch above the silver trim.

TI-99/4a Memory Map

| LED# | ADDRESS | DESCRIPTION |
|---|---|---|
| 1 | >0000->1FFF | Console ROM |
| 2 | >2000->3FFF | Low Exp Mem |
| 3 | >4000->5FFF | DSR Space |
| 4 | >6000->7FFF | Cart. Mem |
| 5 | >8000->9FFF | Console Ram |
| | | Sound Chip |
| | | VDP R/W |
| | | Speech R/W |
| | | GROM R/W |
| 6 | >A000->BFFF | High Exp 1 |
| 7 | >C000->D000 | High Exp 2 |
| 8 | >E000->FFFF | High Exp 3 |

ENJOY!




If you have any questions :

Stephen J. Tuorto
18-Chimney Lane
Bayshore, New York 11706

TIGERCUB SOFTWARE

Over 130 original entertainment, education and programming utility programs in Basic and Extended Basic on cassette or disk. Only $3.00 each! Eighteen different full-disk collections, just $12.00 each! Descriptive catalogue $1.00 refundable. TIPS FROM TIGERCUB full-disk collections of 50+ programs and files from Tigercub Tips newsletters, Vol. I, II and III $15.00 each; any two $27.00; all three $35.00 postpaid. NUTS & BOLTS (#1) and (#2) full disks of 100+ utility subroutines in XBasic 'merge' format, ready to merge into your own programs: $19.95 each, both for $37.00 with documentation, postpaid. Orders to Tigercub Software, 156 Collingwood Ave., Columbus, OH 43213

SMART MODEMS FOR ONLY:
$39.00

Completely TI compatible. Add $6 if using CorComp RS232. Modems are auto dial/answer, come with RS232 cable, manual and telephone cable. Power supply is NOT included. Power requirements are fully documented. Power supplies may be available for a short period of time.

Write to: Smart Modems
14436 FENTON
REDFORD, MI 48239

Send US funds for foreign equivilent. Make check or money order payable to: Scott A Sorel. If a power supply is desired add $10. Money will be refunded if quantities are sold out upon arrival of order.

WANTED: TI DISK DRIVE CONTROLLER CARDS Have you or your friends upgraded to a Myarc, CorComp or Atronic DS/DD Disk Drive Controller Card? We would like to purchase your old TI controller card. We only require the card - and the Disk Manager II module if you still have it. No cables, stand alone units or power supplies. Contact Ryte Data (705) 457-2774 9-5 M-F.

YOUNGTREE

GENEVE COMPUTER $450.00
w/ Enhanced Keyboard $475.00
32k Stand Alone Memory $89.95
99AT Expansion PE Box $155.00
Draw N Plot $12.95
Add $3.00 shipping handling
M/C VISA call 1-216-399-5463
Add $10.00 shipping for PE Box
YOUNGTREE
PO BOX 197
LEAVITTSBURG, OH 44430
Catalog sent with order or send $2.00 (refunded with 1st order)

FOR SALE: COMPLETE SYSTEM.

TI 99/4A Console, Peripheral Expansion Box, 1 Disk Drive w/ controller, 32k Memory Card, RS-232 Card, Joysticks, TI Writer, Extended Basic, Speech Synthesizer, Parsec, TI Invaders etc.)

Contact: C.P. Floyd
26 Chant Cresc.
Unionville, Ontario L3R 1Y8
(416) 477-3684

WANTED: TI 99/4A Peripheral Expansion Cards - 32k, RS232, etc. at used prices. Write: Eric Lilius c/o Ryte Data, 210 Mountain St., Haliburton, Ont. K0M 1S0

Include description and price requested for item. See listing above.

P-CODE SYSTEM FOR SALE:

Full TI P-Code System for TI 99/4A with all Pascal software.

Price: $150.00 (Canadian funds)

Contact: Steve Maly
King's Edgehill School
Windsor, Nova Scotia
B0N 2T0

**R/D COMPUTING** is published monthly by Ryte Data in Haliburton, Ontario. Copyright 1985.

All material is from sources believed to be accurate. The publisher takes no responsibility for errors, omissions or misprints. Articles may be reprinted with credits giving source and address by users groups for publication.

Articles dealing with Texas Instruments 99/4A and 9900 based computers are published. Special attention is given to projects and information on upgrading and modifying the 99/4A console and system.

Please send any information on products, hardware, software, or modifications to our address. We cannot accept responsibility for materials submitted and, unless stated otherwise, will assign all manuscripts etc. for publication.

**Only** manuscripts with sufficient postage and self-addressed mailer will be returned.

Un-classified ad rates are $1.25 per 40 character line. Count ALL characters, spaces, etc. and submit with payment to our address. Ads are run in order received. Deadline is 21st of each month.

**YOU MUST NOTIFY US OF YOUR CHANGE OF ADDRESS SIX WEEKS IN ADVANCE. PLEASE SEND A CHANGE OF ADDRESS CARD WITH YOUR OLD AND NEW ADDRESS.**

## SUBSCRIPTION COST:

$14.00 (US) for 12 issues $20.00 Cdn. all via First Class Mail.
$19.00 (US) overseas Airmail Delivery.

## BACK ISSUES:

Back issues are available for $2.00 each, **subscribers** only for these issues —

**V 2** sold out.
**V 3** TMS 9995 Memory Map & specs numeric keypad project.
**V 4** XB II **plus** by Mechatronic, Myarc 128k card, Autofire joystick project.
**V 5** 32k internal memory project, DS/DD Ramdisk, Auto power-up project.
**V 6** Myarc 256k computer, "C" compiler, RAM/GRAM card.
**V 7** Maximem review, EEROM programmer, Sense and control card.
**V 8** Sold out.
**V 9** Dual disk controller project, RESET switch project.
**V 10-11** Double Issue: LA TI FAIR, Bill Gronos on Assembly, Super load switch, EPROM programmer.
**V12.5** Myarc 640k Geneve, Console speed upgrade, Video monitor filter, Gramkracker review, ATRONIC products.
**V14.5** Expansion Box 8k 'module,' Super Clock Support, Basic Compiler, 9938 Video Chip.
**V15** Video upgrade @ 15mHz, Quad. Density TI Disk Controller, 80 Column Display preview.
**V16** P Box Modification, Multi-Module project.
**V17** Tigercub, Eprommer update, Quad density update.
**V18** Power Supply, RAVE 99 Keyboard, Year Review.
**V19** Geneve, c993.0, RAM Disk backup, Hi-Speed cassette.
**V20** 99AT, T.O.D., Horizon modification, Rapid Copy 32k project, GRAM card software, etc.
**V21** Expansion system, 8k DSR RAM Project, C99 by R. Albright, TI Writer in Memory, AE1 etc.
**V22** EEPROMS, C99 part II, File Utiliter, Command DOS, 8k DSR RAM part II.
**V23** Hardware group, co-processors, Improved Video, Real Time Clock, C99 part III, Memory LED's.
**V24** Triple Tech project, 80 Column Display II, RGB Conversion Project Part II, 64k
**V25** German GRAM CRACKER, Co-processor update, Proto Board, Console Calc., etc. 64k on the bus project.

Ryte Data
210 MOUNTAIN STREET
HALIBURTON, ONTARIO K0M 1S0
705 457-2774